Loading N-Dimensional Vector into Quantum Registers from Classical Memory with O(logN) Steps*

Pang Chao-Yang†

[1] *Key Software Laboratory, Sichuan Normal University, Chengdu 610066, China*

[2] *College of Mathematics and Software Science, Sichuan Normal University, Chengdu 610066, China*

Abstract

Vector is the general format of input data of most algorithms. Designing unitary operation to load all information of vector into quantum registers of quantum CPU from classical memory is called quantum loading scheme (QLS). QLS assembles classical memory and quantum CPU as a whole computer, which will be important for further quantum computation. We present a QLS based on path interference with time complexity $O(log_2 N)$, while classical loading scheme has time complexity $O(N)$, that is the efficiency bottleneck of classical computer.

Keywords: Path Interference, Entangled State, Quantum Loading Scheme

PACC: 4230, 4230N, 4230V

*The work is supported by the key item of science and technology awarded by Science and Technology Bureau of Sichuan Province under Grant No. 2006J13-153, and the key item awarded by Sichuan Normal Univ. under Grant No.06lk002

†Electronic address: cyp˙900@hotmail.com; Electronic address: cypang@sicnu.edu.cn

I. INTRODUCTION

Vector is the general format of input data of algorithm, and each component of vector is stored in classical memory sequentially in general. Thus there are two questions for general quantum algorithm, the first question is that which state is suitable to represent all information of vector for further quantum computation, and the second question is that how to load all information of vector into quantum registers (or quantum state) without losing information. Loading data set such as vector into classical registers of CPU from classical memory is called **classical loading scheme (CLS)**. Similar to CLS, designing unitary operation to load all information of vector into quantum registers of quantum CPU from classical memory is called **quantum loading scheme (QLS)**. CLS or QLS assembles classical memory and CPU as a whole computer. QLS makes quantum CPU is compatible with classical memory.

An $N-$dimensional vector is denoted as $\vec{a} = \{a_0, a_1, ..., a_{N-1}\}$, where the components $a_0, a_1, ..., a_{N-1}$ are real numbers. It has been shown that entangled state $\frac{1}{\sqrt{N}}(\sum\limits_{i=0}^{N-1} |i\rangle_{register1}|a_i\rangle_{register2})$ is suitable for the representation of vector without losing any information of vector [1, 2, 3, 4]. Let initial state $|\phi_0\rangle$ be $|\phi_0\rangle = |0\rangle_{q_1q_2...q_n}|0\rangle_{p_1p_2...p_m}|ancilla_1\rangle$, where the $m+n$ qubits $q_1, ..., q_n, p_1, ..., p_m$ are collected as a whole object, dividing them is prohibited, and where the ancillary state $|ancilla_1\rangle$ is *known*. QLS can be described as how to design unitary operation $U_{(0,1,...,N-1)}$ such that

$$|\phi\rangle = U_{(0,1,...,N-1)}|\phi_0\rangle = \frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1} |i\rangle_{q_1q_2...q_n}|a_i\rangle_{p_1p_2...p_m})|ancilla_2\rangle, \tag{1}$$

where $N = 2^n$ and the ancillary state $|ancilla_2\rangle$ is *known* (all ancillary states is *known* in this paper.).

Nielsen and Chuang pointed out that quantum computer should have loading scheme in principle to load classical database record into quantum registers from classical database [6, 7, section 6.5]. However, there is no detailed work on QLS up till now. In fact, the research of QLS is motivated by the quantum algorithm of image compression [1, 2, 3, 4, 5]. In this paper, we present a QLS based on the path interference, which has been widely used in quantum information processing, e.g. non-unitary computation[8, 9]. The unitary computation using path interference is demonstrated in this paper, and the output of the unitary computation can be measured with successful probability 100% in theory. The

time complexity of our QLS is $O(log_2N)$, which exhibits a speed-up over CLS with time complexity $O(N)$.

II. THE DESIGN OF QLS

A. Loading 2D Vector into Quantum Registers from Classical Memory

The design of unitary operation $U_{(0,1)}$ that loads 2D vector $\vec{a} = \{a_0, a_1\}$ is described conceptually as follows (see Fig.1):

Step 1 The switch S_1 applies rotation on the initial ancilla state and transforms $|Off_0\rangle$ into

$$|Off_0\rangle \stackrel{S_1}{\rightarrow} \frac{|Off_1\rangle + |On_1\rangle}{\sqrt{2}}$$

and generate the following state $|\phi_1\rangle$

$$|\phi_1\rangle = \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|0\rangle_{p_1p_2\cdots p_m}|On_1\rangle + \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|0\rangle_{p_1p_2\cdots p_m}|Off_1\rangle \tag{2}$$

Step 2 Perform unitary operations I_0 and A_0 along 'On_1' path, while perform unitary operations I_1 and A_1 along 'Off_1' path.

$$\begin{cases} |0\rangle_{q_1q_2...q_n} \stackrel{I_0}{\rightarrow} |0\rangle_{q_1q_2...q_n} \\ |0\rangle_{p_1p_2\cdots p_m} \stackrel{A_0}{\rightarrow} |a_0\rangle_{p_1p_2\cdots p_m} \end{cases}, \begin{cases} |0\rangle_{q_1q_2...q_n} \stackrel{I_1}{\rightarrow} |1\rangle_{q_1q_2...q_n} \\ |0\rangle_{p_1p_2\cdots p_m} \stackrel{A_1}{\rightarrow} |a_1\rangle_{p_1p_2\cdots p_m} \end{cases}$$

We assume the output of two pathes are simultaneous, then the state $|\phi_2\rangle$ is generated as

$$\begin{cases} \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|0\rangle_{p_1p_2\cdots p_m}|On_1\rangle \stackrel{A_0I_0}{\rightarrow} \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|a_0\rangle_{p_1p_2\cdots p_m}|On_1\rangle \\ \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|0\rangle_{p_1p_2\cdots p_m}|Off_1\rangle \stackrel{A_1I_1}{\rightarrow} \frac{1}{\sqrt{2}}|1\rangle_{q_1q_2...q_n}|a_1\rangle_{p_1p_2\cdots p_m}|Off_1\rangle \end{cases}$$

$$\Rightarrow |\phi_2\rangle = \frac{1}{\sqrt{2}}|0\rangle_{q_1q_2...q_n}|a_0\rangle_{p_1p_2\cdots p_m}|On_1\rangle + \frac{1}{\sqrt{2}}|1\rangle_{q_1q_2...q_n}|a_1\rangle_{p_1p_2\cdots p_m}|Off_1\rangle \tag{3}$$

The functions of I_0 and I_1 are to generate subscripts 0 and 1 respectively, and the functions of A_0 and A_1 are to generate numbers a_0 and a_1 respectively. Because value a_0 and a_1 are both known numbers, flipping part of the $m+n$ qubits $q_1, ..., q_n, p_1, ..., p_m$ will generate states $|a_0\rangle_{p_1p_2\cdots p_m}$ or $|a_1\rangle_{p_1p_2\cdots p_m}$. Thus, the unitary operations I_0, I_1, A_0, A_1 is easy to be designed.

Step 3 The switch S_2 applies rotation on the initial ancilla state as

$$\begin{cases} |Off_1\rangle \xrightarrow{S_2} \frac{|Off_2\rangle - |On_2\rangle}{\sqrt{2}} \\ |On_1\rangle \xrightarrow{S_2} \frac{|Off_2\rangle + |On_2\rangle}{\sqrt{2}} \end{cases}$$

and generate the following state $|\phi_3\rangle$

$$\begin{aligned} |\phi_3\rangle \;=\; & \frac{1}{2}(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\cdots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\cdots p_m})|Off_2\rangle \\ & +\frac{1}{2}(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\cdots p_m} - |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\cdots p_m})|On_2\rangle \end{aligned} \quad (4)$$

Step 4 Apply phase transformaition B along 'On_2' path.

$$B = |0\rangle\,|a_0\rangle\langle a_0|\langle 0| - |1\rangle\,|a_1\rangle\langle a_1|\langle 1| \quad (5)$$

It's a very fast operation and generates the state $|\phi_4\rangle$

$$|\phi_4\rangle = \frac{1}{\sqrt{2}}(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\cdots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\cdots p_m})(\frac{|Off_2\rangle + |On_2\rangle}{\sqrt{2}}) \quad (6)$$

Step 5 The switch S_3 applies rotation on the initial ancilla state as

$$\begin{cases} |Off_2\rangle \xrightarrow{S_3} \frac{|Off_3\rangle + |On_3\rangle}{\sqrt{2}} \\ |On_2\rangle \xrightarrow{S_3} \frac{|Off_3\rangle - |On_3\rangle}{\sqrt{2}} \end{cases}$$

and generate the final state $|\phi\rangle$

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|0\rangle_{q_1q_2\ldots q_n}|a_0\rangle_{p_1p_2\cdots p_m} + |1\rangle_{q_1q_2\ldots q_n}|a_1\rangle_{p_1p_2\cdots p_m})|Off_3\rangle \quad (7)$$

Fig.1 and Eq.(8) illustrate the processing of operation $U_{(0,1)}$.

$$\begin{aligned} & |0\rangle|0\rangle|Off_0\rangle \xrightarrow{S_1} \left\langle \begin{array}{c} \frac{1}{\sqrt{2}}|0\rangle|0\rangle|On_1\rangle \xrightarrow{A_0I_0} \frac{1}{\sqrt{2}}|0\rangle|a_0\rangle|On_1\rangle \\ \frac{1}{\sqrt{2}}|0\rangle|0\rangle|Off_1\rangle \xrightarrow{A_1I_1} \frac{1}{\sqrt{2}}|1\rangle|a_1\rangle|Off_1\rangle \end{array} \right\rangle \xrightarrow{S_2} \\ & \left\langle \begin{array}{c} \frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|Off_2\rangle \\ \frac{1}{2}(|0\rangle|a_0\rangle - |1\rangle|a_1\rangle)|On2\rangle \xrightarrow{B} \frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|On_2\rangle \end{array} \right\rangle \xrightarrow{S_3} |\phi\rangle \end{aligned} \quad (8)$$

Here as well as in the following discussions, all subscripts of registers are ignored.

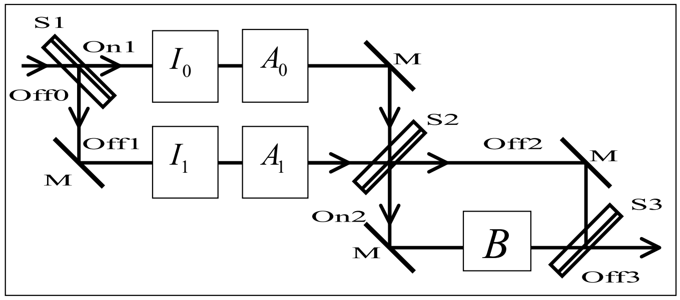


FIG. 1: **The Illustration of the processing of unitary operation $U_{(0,1)}$ that transforms state from $|0\rangle\,|0\rangle\,|Off_0\rangle$ into $\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)\,|Off_3\rangle$.** M: Mirror

B. Loading 4D Vector or Multi-Dimensional into Quantum Registers

The design of unitary operation $U_{(0,1,2,3)}$ is described conceptually as follows (see Fig.2):

Step 1 Construct unitary operation S_4, S_5, S_6, B' as following:

$$\begin{cases} |Off_i\rangle \xrightarrow{S_i} \frac{|Off_{i+1}\rangle+|On_{i+1}\rangle}{\sqrt{2}} \\ |On_i\rangle \xrightarrow{S_i} \frac{|Off_{i+1}\rangle-|On_{i+1}\rangle}{\sqrt{2}} \end{cases}, \begin{cases} |Off_5\rangle \xrightarrow{S_5} \frac{|Off_{i+1}\rangle-|On_{i+1}\rangle}{\sqrt{2}} \\ |On_5\rangle \xrightarrow{S_5} \frac{|Off_{i+1}\rangle+|On_{i+1}\rangle}{\sqrt{2}} \end{cases}, \quad B' = |\alpha\rangle\langle\alpha| - |\beta\rangle\langle\beta|,$$

where $i = 4, 6$, $|\alpha\rangle = \frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)$ and $|\beta\rangle = \frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)$

Step 2 Assemble unitary operations S_4, S_5, $S_6, B', U_{(0,1)}$ and $U_{(2,3)}$ according to Fig.2 to form unitary operations $U_{(0,1,2,3)}$.

Eq.(9) illustrates the processing of operation $U_{(0,1,2,3)}$.

$$\begin{array}{c} |0\rangle|0\rangle|Off_4\rangle \xrightarrow{S_4} \left\langle \begin{array}{c} \frac{1}{\sqrt{2}}|0\rangle|0\rangle|On_5\rangle \xrightarrow{U_{(0,1)}} \frac{1}{2}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle)|On_5\rangle \\ \frac{1}{\sqrt{2}}|0\rangle|0\rangle|Off_5\rangle \xrightarrow{U_{(2,3)}} \frac{1}{2}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)|Off_5\rangle \end{array} \right\rangle \xrightarrow{S_5} \\ \left\langle \begin{array}{c} \frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle) + \frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)]|Off_6\rangle \\ \\ \frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle) - \frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)]|On_6\rangle \xrightarrow{B'} \\ \frac{1}{2}[\frac{1}{\sqrt{2}}(|0\rangle|a_0\rangle + |1\rangle|a_1\rangle) + \frac{1}{\sqrt{2}}(|2\rangle|a_2\rangle + |3\rangle|a_3\rangle)]|On_6\rangle \end{array} \right\rangle \xrightarrow{S_6} |\phi\rangle \end{array} \tag{9}$$

If the unitary operations $U_{(0,1)}$ and $U_{(2,3)}$ embedded in Fig.2 are replaced by $U_{(0,1,2,3)}$ and $U_{(4,5,6,7)}$ respectively, then $U_{(0,1,\ldots,7)}$ is constructed. Similar to Fig.2, we can apply the same method to construct unitary operation $U_{(0,1,\ldots,2^n)}$. If $N \neq 2^n$, we could add extra zero components to create a 2^n-dimensional vector.

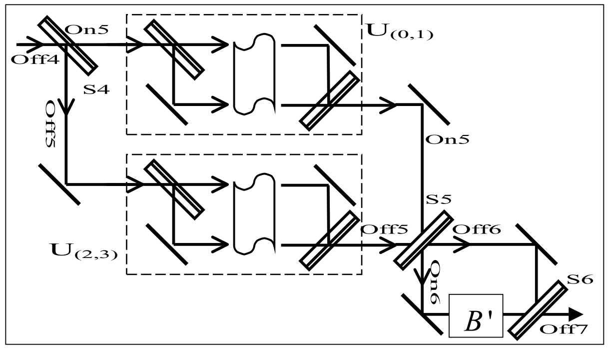


FIG. 2: **The illustration of unitary operation $U_{(0,1,2,3)}$ that transforms state from $|0\rangle\,|0\rangle\,|Off_4\rangle$ into $(\sum\limits_{i=0}^{3}\frac{1}{2}\,|i\rangle\,|a_i\rangle)\,|Off_7\rangle$.**

C. Loading Vector into State $\frac{1}{\sqrt{N}}(\sum\limits_{i=0}^{N-1}|i\rangle\,|0\rangle)$ to Form Entangled State $\frac{1}{\sqrt{N}}(\sum\limits_{i=0}^{N-1}|i\rangle\,|a_i\rangle)$

Grover's algorithm [6] has the function that find the index i_0 of a special database record $record_{i_0}$ from the index superposition of state $\frac{1}{\sqrt{N}}(\sum\limits_{i=0}^{N-1}|i\rangle)$ taking $O(\sqrt{N})$ steps. And the record $record_{i_0}$ is the genuine answer wanted by us. However, the corresponding record $record_{i_0}$ can not be measured out unless the 1-1mapping relationship between index i and the corresponding record $record_i$ is bound in the entangled state $\frac{1}{\sqrt{N}}(\sum\limits_{i=0}^{N-1}|i\rangle\,|record_i\rangle)$. That is, we need a unitary operation U_L such that

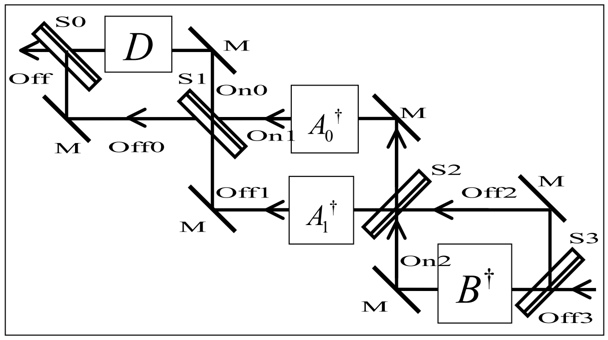
$$\frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle\,|0\rangle)\,|ancilla_4\rangle \xrightarrow{U_L} \frac{1}{\sqrt{N}}(\sum_{i=0}^{N-1}|i\rangle\,|a_i\rangle)\,|ancilla_3\rangle \tag{10}$$

Ref.[3, 4] generalize Grover's algorithm to the general search case with complex computation, and U_L is required in this general search case.

U_L can be designed using the same method shown in Fig.1 and Fig.2. Fig.3 shows the design of the inverse unitary operation $(U_L)^\dagger$ at the case $N=2$. U_L has time complexity $O(log_2 N)$ (unit time: phase transformation and flipping the qubits of registers).

It has been demonstrated that giant molecules, such as charcoal c_{60}, exhibit quantum interference [10]. Thus many freedom degrees of giant molecule can be regarded as qubits to realize the QLS presented in this paper. In addition, one of QLS application is that QLS can load the data of image with huge size into quantum registers at a time for further image compression [1, 2, 3, 4], while only one data can be loaded into registers at a time for classical computer.

FIG. 3: **The Illustration of Unitary Operation** $(U_L)^\dagger$: $\frac{1}{\sqrt{2}}(\sum_{i=0}^{1}|i\rangle|a_i\rangle)|Off_3\rangle \rightarrow \frac{1}{\sqrt{2}}(\sum_{i=0}^{1}|i\rangle|0\rangle)|Off\rangle$. Operation U_L can be designed using the same method shown in Fig.1 and Fig.2. S_0: $|Off_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|Off\rangle + |On\rangle)$, $|On_0\rangle \rightarrow \frac{1}{\sqrt{2}}(|Off\rangle - |On\rangle)$. Phase transformation $D = |i_1\rangle|0\rangle\langle 0|\langle i_1| - |i_0\rangle|0\rangle\langle 0|\langle i_0|$, where $i_0 = 0$, $i_1 = 1$.



III. CONCLUSION

Designing simple and fast unitary operation to load classical data set, such as vector, into quantum registers from classical memory is called quantum loading scheme (QLS). QLS makes quantum CPU is compatible with classical memory, and it assembles classical memory and quantum CPU as a whole. QLS is the base of further quantum computation. The QLS with time complexity $O(log_2N)$ (unit time: phase transformation and flipping the qubits of registers) is presented in this paper, while classical loading scheme (CLS) has time complexity $O(N)$ (unit: addition) because all computation instructions have to be executed one by one. Path interference is applied to design QLS in this paper so that the complexity of designing quantum algorithm is decomposed as the design of many simple unitary operations. In addition, this paper demonstrates that using path interference to design unitary operation and parallel quantum computation is possible.

Acknowledgments

The author thanks Dr. Z.-W Zhou who is at Key Lab. of Quantum Information, Uni. of Science and Technology of China for that he points out two errors in author's primary idea. The author's first error is that the result generate with probability 50% for 2D vector, the second error is the defect of Fig.2 that the output is direct product state. Dr. Z.-W

Zhou tries his best to help author for nearly 3 years. The author thanks his teacher, prof. G.-C Guo. The author is brought up from Guo's Lab.. The author thanks prof. V. N. Gorbachev who is at St.-Petersburg State Uni. of Aerospace Instrumentation for the useful discussion with him. The author thanks Mir. N. Kiesel who is at Max-Plank-Institute fur Quantenoptik, Germany for his checking the partial deduction of section 2 of this paper. The author thanks prof. G.-L Long who is at Tsinghua Uni., China for the useful discussion with him and the author obtains some heuristic help from his eprint file quant-ph/0512120. The author thanks associate prof. Shiuan-Huei Lin who is at National Chiao Tung Uni., Taiwan., China for encouraging the author. The author thanks prof. Hideaki Matsueda who is at Kochi Uni., Japan for encouraging the author. The author thanks prof. J. Zhang and B.-P Hou who are at Sichuan Normal Uni., China for their help. The author thanks prof. Z.-F. Han, Dr. Y.-S. Zhang, Dr. Y.-F. Huang, Dr. Y.-J. Han, Mr. J.-M Cai, Mr. M.-Y. Ye, and Mr. M.-Gong for their help and suggestions. One of reviewers presents many significative suggestions to improve the readability of this paper, the author thanks the reviewer.

---

[1] Pang C Y 2006 Quantum Image Compression (Postdoctoral Report) (Heifei: University of Science and Technology of China)

[2] Pang C Y, Zhou Z W, and Guo G C 2006 *Chin. Phys.* **15** 3039

[3] Pang C Y, Zhou Z W, and Guo G C 2006 arXiv:quant-ph/0601043

[4] Pang C Y, Zhou Z W, Chen P X, and Guo G C 2006 *Chin. Phys.* **15** 618

[5] Latorre J I 2005 arXiv:quant-ph/0510031

[6] Nielsen M A and Chuang I L 2000 Quantum Computationand and Quantum Information (London: Cambridge University Press)

[7] Zhao Q C(translator), Nielsen M A and Chuang I L 2004 Quantum Computation and Quantum Information (Beijing: Tsinghua University publishers) (in Chinese)

[8] Hosten O, Rakher M T, Barreiro J T, Peters N A, Kwiat P G 2006 *Nature* **439** 949

[9] Long G L 2006 *Communications in Theoretical Physics* **45** 825 .

[10] Arndt M, Nairz O, Vos-Andreae J, Keller C, van der Zouw G, and Zeilinger A. 1999 *Nature*(London), **401** 680